(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 November 2002 (21.11.2002)

PCT

(10) International Publication Number
WO 02/093863 A2

(51) International Patent Classification⁷: H04L 29/00

(21) International Application Number: PCT/EP02/05230

(22) International Filing Date: 13 May 2002 (13.05.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
01111706.6 14 May 2001 (14.05.2001) EP

(71) Applicant (*for all designated States except US*): **TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; S-126 25 Stockholm (SE).

(72) Inventors; and
(75) Inventors/Applicants (*for US only*): **FEYERABEND, Konrad** [DE/SE]; Akerhielmsgatan 27, SE-167 33 Bromma (SE). **MC AULEY, John** [IE/IE]; 194 Athlumney Castle, Meath, Navan (IE). **POLLE, Torsten** [DE/DE]; Tappenstrasse 40, 31134 Hildesheim (DE).

(74) Agent: **MOHSLER, Gabriele**; Ericsson Eurolab Deutschland GmbH, Ericsson Allee 1, 52134 Herzogenrath (DE).

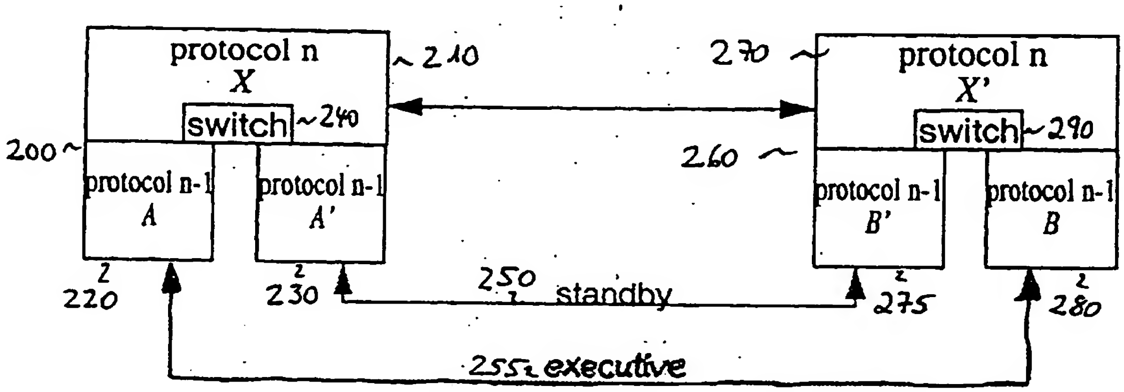
(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW, ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent*

[Continued on next page]

(54) Title: APPLICATION TRANSPARENT REDUNDANCY IN A PROTOCOL STACK



(57) Abstract: The invention relates to a method in a communication network, wherein a first peer is provided for a communication with a second peer by means of a protocol stack comprising a first protocol layer and a second protocol layer provided at the first peer, said second protocol layer having a second layer instance, wherein the protocol layers are provided for communication with corresponding protocol layers of a further peer, and wherein the first protocol layer comprises an executive protocol instance assigned to a first protocol layer communication link and standby protocol instance assigned to a further first protocol layer communication link, wherein the executive protocol instance performs first protocol layer tasks, and wherein the standby protocol instance is kept in a standby mode for taking over the first protocol layer tasks form the executive protocol instance. The invention relates further to a protocol stack, network element and computer program.

WO 02/093863 A2

(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Application transparent redundancy in a protocol stack

The invention addresses the field of protocol-based communication systems. It relates to a method provided for communication using a redundant protocol layer, a protocol stack, network element, communication system and computer program, in particular for use in a telecommunication network.

Background of the invention

When two peers of a communication system communicate with each other, they use a protocol, which defines the format and sequence of messages that are exchanged. Typically the protocol is based on a lower protocol, i.e. a protocol on a lower protocol layer. The definition of lower and higher protocol layers follows from the well-known ISO/OSI reference model. Fig. 1 shows an example of such known protocol stacks 100, 160. A higher layer protocol n 110, 160 is based on a lower layer protocol n-1 120, 150. A failure, either a hardware failure or a software failure, in the lower protocol layers 110, 160 will block the communication between the peers 100, 160 on the higher protocol layer 110, 160.

Therefore, it is an object of the invention to provide a method, a protocol stack, a network element, a communication system and a computer program that protect a layered protocol stack against hardware and / or software failures occurring in a protocol layer, in particular in one or more lower protocol layers.

Summary of the invention

This object is solved by the teaching of the independent claims.

The invention relates to a method in a communication network, wherein a first peer is provided for communication with at least a second peer by means of a protocol stack that is provided at the first peer, said protocol stack comprising a first protocol layer and a second protocol layer, said second protocol layer having a second protocol layer instance, wherein each of the protocol layers are provided for

communication with a corresponding protocol layer of at least one further peer. The first protocol layer comprises an executive protocol layer instance assigned to a first protocol layer communication link and at least one standby protocol layer instance assigned to a further first protocol layer communication link, wherein the executive protocol layer instance performs first protocol layer tasks required for the communication, and wherein the standby protocol layer instance is kept in a standby mode for taking over the first protocol layer tasks from the executive protocol layer instance.

In a further aspect relates the invention to a protocol stack comprising a first protocol layer and a second protocol layer, said second protocol layer having a second protocol layer instance, wherein the first protocol layer comprises an executive protocol layer instance adapted to perform first protocol layer tasks, and at least one standby protocol layer instance adapted to be kept in a standby mode for taking over the first protocol layer tasks from the executive protocol layer instance, and wherein the second protocol layer comprises a switching instance adapted to control which of the first protocol layer instances performs the first protocol layer tasks.

Preferably is the first protocol layer a so-called lower protocol layer, while the second protocol layer is a so-called higher protocol layer according to the ISO/OSI reference model.

Due to the provisioning of executive and standby protocol layer instances is a redundancy introduced into the protocol stack, which increases the reliability of communication nodes or systems using the corresponding protocol stacks. This is in particular useful for communication traffic having real-time requirements, e.g. in telecommunication systems, or being security-relevant, e.g. in control signalling networks of automated industrial production systems.

Advantageously, an operator of a communications system applying the described method can configure in the system, which first layer protocol instance shall be the executive protocol layer instance, and, consequently, which one shall be the standby protocol layer instance. The operator might change the configuration in the event of a detected failure.

As a further advantage, the executive protocol layer instance and the standby protocol layer instance can use different physical communication media, can be hosted by different physical instances and can implement different protocols. Doing so, maximum protection against failures in the first protocol layer can be reached.

Advantageously can the invention be applied transparently, i.e. invisible, for communication parties or communication applications.

In a further aspect relates the invention to a network element of a communication network adapted to perform the method according to the invention. In another aspect relates the invention to a communications network comprising at least one corresponding network element. Such a network element can be, e.g., a network node of a mobile communication system, like a Mobile Switching Centre MSC, a Packet MSC, a Serving General Packet Radio Service Support Node SGSN or a Home Location Register HLR, a Base Station BS, a Base Station Controller BSC, a Radio Network Controller RNC or a Mobile Station MS, e.g. a mobile phone, a Personal Digital Assistant PDA or a laptop computer adapted to connect to the mobile communication network.

In a further aspect relates the invention to a computer program, loadable into an internal memory of a digital processing unit, comprising software code portions adapted to control the steps according to the inventive method, when the computer program is executed on the digital processing unit.

Further embodiments of the invention are described in the dependent claims.

In one embodiment of the method comprises the protocol stack a switching instance that controls which of the first protocol layer instances performs the first protocol layer tasks. Advantageously, the control instance is located close to the first protocol layer, thus minimising signalling effort and response times.

In a another embodiment monitors the switching instance whether communication data are received via the standby protocol layer instance, and the switching instance assigns the first protocol layer tasks to the standby protocol layer instance, if the monitoring indicates that communication data are received via the standby protocol layer instance. Advantageously, there is no separate signalling required as a trigger for switching to the standby instance. This embodiment can preferably be used in a protocol stack having a slave function with respect to the redundancy mechanism.

In a further embodiment is a switching request received by the first peer, preferably by the switching instance of said peer, and the switching instance assigns responsively the first protocol layer tasks to the standby protocol layer instance. This allows for a switching to the standby instance, even if the protocol stack processes currently no further communication traffic that could otherwise be used as a switching trigger.

In another embodiment performs the switching instance at least once a supervision of the executive protocol layer instance operation, in other words, the whole executive first protocol layer communication link, by initiating the sending of a supervision request via at least one of the first protocol layer instances, and by monitoring whether a corresponding supervision response is received, and assigns the first protocol layer tasks to the standby protocol layer instance if the supervision indicates a failure on the executive protocol layer instance. This embodiment is useful for a protocol stack acting as a master with respect to the set-up of the communication configuration. In particular a supervision of the first layer instances

operation in dedicated time periods allows for a reliable detection of first protocol layer failures.

In a further embodiment sends the switching instance a switching request at least to the second peer, if the supervision indicates a failure on the executive protocol layer instance, said switching request requesting the assignment of the first protocol layer tasks to the standby protocol layer instance. A protocol stack acting as a master in view of the executive/standby-configuration can therefore inform a slave protocol stack without delay about any failure in the first protocol layer. Advantageously, a slave protocol stack has no need to perform an own supervision, but can rely upon the master protocol stack.

In another embodiment is the switching instance located in the second protocol layer.

In one embodiment is the invention realised by software. In a further preferred embodiment is the computer program stored on a computer readable medium like a CD-ROM, a floppy disc, optical disc or hard disk. Therefore, a good physical portability of the software is provided, i.e. upgrades can be performed in an easy way.

Another embodiment of the invention addresses a protocol stack comprising redundant lower layers (executive and standby) and a common higher layer including detection means for detection a link failure and a switching means for switching from the executive lower layer to the standby lower layer. Further embodiments refer to a protocol stack having redundant lower layers, a corresponding network element, a corresponding communications network and a corresponding computer program.

In the following are the invention and its embodiments described in detail with reference to the figures, which show:

Fig. 1	a known protocol layering,
Fig. 2	a protocol layering according to the invention,
Fig. 3	a redundant Intra-Network Packet Data Mobile Application Part PMAP,
Fig. 4	a message flow for a start-up sequence,
Fig. 5	a message flow for choosing the executive protocol layer instance,
Fig. 6	a message flow for a supervision sequence, and
Fig. 7	a message flow for a fail over sequence.

Detailed description of embodiments of the invention

In the embodiment described in the following uses the invention two lower layer protocol instances and builds a switch into a higher layer instance of a protocol stack. The terms 'lower layer' and 'higher layer' are to be understood in accordance to the ISO/OSI reference model. The higher layer instance chooses one of the lower layer instances as executive and one as standby instance. The actual communication is made over the executive lower layer instance.

As shown in Fig. 2, the higher protocol instances 210, 270 of both peers 200, 260 allocate two lower layer protocol instances 220, 230, 275, 280. Those lower layer instances of one peer do not have to use the same physical communication medium, do not have to be hosted by the same physical instance and do not have to implement the same protocol.

The pair of instances that consists of the two executive instances is called the executive link. The other pair is called the standby link.

If a connection-oriented protocol is used in the lower layer protocol, connections between the corresponding lower layer protocol are established. That is, protocol instance A 220 and A' 230 establish connections 255, 250 to protocol instances B 280 and B' 275, respectively.

If a connectionless lower layer protocol is used, the protocol instances A 220, A' 230 and B 280, B' 275 have to be prepared to transfer data between the respective instance according to the requirements of the lower layer.

A switch component 240, 290 decides which of the lower layer instances acts as executive lower layer instance and which as standby lower layer instance. Both switch instances in the peers have to make the same decision. In order to achieve this unique decision, one of the higher layer protocol instance switches (X 200 or X' 260) becomes the master while the other peer switch becomes the slave. The master selects the executive link at its discretion and notifies the slave, e.g. using the executive link.

In the event of a failure on the executive link, the standby link is activated and made the executive link. Failure detection can be achieved by using services provided by the lower layers themselves or by supervision from the higher layers.

The switch component transparently routes messages from/to the higher layer peers over the executive link.

In the following is with reference to Fig. 3 as an example an embodiment for use of the invention in a telecommunication system comprising two different nodes described. Corresponding message flows are described with reference to the figures 4 to 7, in which same entities are provided with same reference signs.

Intra-Network Packet Data Mobile Application Part PMAP is the protocol used to exchange control-signalling information between two different types of nodes V 360 and P 300. PMAP uses Transmission Control Protocol / Internet Protocol TCP/IP as lower layer protocol. Nodes of type V 360 have two physical devices that terminate TCP/IP connections over Ethernet. The invention allows exploiting this physical redundancy but still hiding the redundancy from the node applications.

When a node of type V 360 starts up, the node launches the PMAP protocol instance V 370; 425. The PMAP protocol instance V 370; 425 starts two TCP/IP protocol instances 375, 380; 415, 420 in server mode, i.e. instances that wait for incoming connection requests. Both instances B 380; 420 and B' 375; 415 are started each on a separate physical device. A corresponding message flow of the start-up sequence is shown in Fig. 4. The start of the server mode TCP/IP protocol instances 420, 415 is triggered by corresponding 'create' messages 430, 435.

When a node of type P 300 starts up, the node launches the PMAP protocol instance P 310; 400. The PMAP protocol instance P310; 400 starts two TCP/IP protocol instances 320, 330; 405, 410 in client mode, i.e. instances that try to connect to servers. Both instances A 320; 405 and A' 330; 410 are started on the same physical device.

Referring to the message flow of Fig. 4, the launching of the TCP/IP protocol instances 320, 330; 405, 410 is triggered by corresponding 'create' messages 440, 455 that indicate the destination servers 420; 415, which are targeted by a 'connect' message 445, 460 each. The servers 420, 415 notify the protocol instance V 425 about the successful start-up by means of 'established' messages 450, 465.

Referring to Fig. 3, the PMAP protocol instance P 310 acts as master while the PMAP protocol instance V 370 acts as slave. That is, the PMAP protocol instance P 310 notifies instance V 370 of its decision by sending a specific message over the chosen executive link. Since messages delivery and order is guaranteed by TCP/IP both PMAP protocol instances 310, 370 regard the same link as the executive link 355.

A message flow that illustrates the signalling of the choice of the master is shown in Fig. 5. PMAP protocol instance P 310; 400 transmits a 'send' message 510 to the client A 320; 405, which forwards the message further 520 to the server B 380; 420.

Both PMAP protocol instances 310, 370; 400, 425 notify their users 500, 505 of the successful establishment of a data 'logical' link.

Referring to Fig. 3, the master, i.e. PMAP protocol instance P 310, sends out supervision requests over both links at regular intervals. The slave, i.e. PMAP protocol instance V 370 has to answer those requests on the respective links within a certain period of time. If the master 310 fails to receive a supervision reply within a certain period of time, the master 310 considers the link as broken. If the link was the executive link 355 the master 310 tries to activate the standby link 350 as executive link. A corresponding message flow for a supervision without a detection of a failure is illustrated in Fig. 6, while Fig. 7 shows a scenario, wherein the supervision request times out 730, and therefore a change of the executive protocol instance is signalled 740, 750, 760 via the standby link 355.

Without being limited to, the invention can be used preferably in all communications systems as defined by 3GPP in the release 99 as well as in all future releases thereof. In particular, this includes Personal Digital Cellular PDC systems and Packet Personal Digital Cellular PPDC systems as well as Universal Mobile Telecommunication Service UMTS networks, core networks, Wireless Local Area Networks WLAN, GSM networks, Internet Protocol IP based data systems, as well as the corresponding user equipment.

Claims

1. Method in a communication network, wherein a first peer (200; 300) is provided for communication with at least a second peer (260; 360) by means of a protocol stack that is provided at the first peer (200; 300), said protocol stack comprising a first protocol layer and a second protocol layer, said second protocol layer having a second protocol layer instance (210; 310; 400), wherein each of the protocol layers is provided for communication with a corresponding protocol layer of at least one further peer,

characterised in that

the first protocol layer comprises an executive protocol layer instance (220; 320) assigned to a first protocol layer communication link (255; 355) and at least one standby protocol layer instance (230; 330) assigned to a further first protocol layer communication link (250; 350), wherein the executive protocol layer instance (220; 320) performs first protocol layer tasks required for the communication, and wherein the standby protocol layer instance (230; 330) is kept in a standby mode for taking over the first protocol layer tasks from the executive protocol layer instance (220; 320).

2. Method according to claim 1, wherein the protocol stack comprises a switching instance (240; 340) that controls which of the first protocol layer instances (220, 230; 320, 330) performs the first protocol layer tasks.

3. Method according to claim 2, wherein the switching instance (240; 340) monitors whether communication data are received via the standby protocol layer instance (230; 330), and wherein the switching instance (240; 340) assigns the first protocol layer tasks to the standby protocol layer instance (230; 330), if the monitoring indicates that communication data are received via the standby protocol layer instance (230; 330).

4. Method according to claim 2 or 3, wherein a switching request is received by the first peer (200; 300), and wherein the switching instance (240; 340) responsively

assigns the first protocol layer tasks to the standby protocol layer instance (230; 330).

5. Method according to any of the claims 2 to 4, wherein the switching instance (240; 340) performs at least once a supervision of the executive protocol layer instance operation by initiating the sending of a supervision request (600, 610; 700, 710) via at least one of the first protocol layer instances (220, 230; 320; 330), and by monitoring whether a corresponding supervision response (650) is received, and assigns the first protocol layer tasks to the standby protocol layer instance (330; 375) if the supervision indicates a failure (730) in the executive protocol layer instance operation.

6. Method according to claim 5, wherein the switching instance (240; 340) sends a switching request (740; 750) at least to the second peer, if the supervision indicates a failure (730) on the executive protocol layer instance operation, said switching request requesting the assignment of the first protocol layer tasks to the standby protocol layer instance (275; 375).

7. Method according to any of the claims 2 to 6, wherein the switching instance (240; 340) is located in the second protocol layer (210; 310).

8. Protocol stack comprising a first protocol layer and a second protocol layer, said second protocol layer having a second protocol layer instance (210; 310)
<u>characterized in that</u>
the first protocol layer comprises an executive protocol layer instance (220; 320) adapted to perform first protocol layer tasks, and at least one standby protocol layer instance (230; 330) adapted to be kept in a standby mode for taking over the first protocol layer tasks from the executive protocol layer instance (220; 320), and that the second protocol layer comprises a switching instance (240; 340) adapted to control which of the first protocol layer instances (220,230; 320, 330) performs the first protocol layer tasks.

9. Network element of a communication network adapted to perform the method according to any of the claims 1 to 7.

10. Communications network comprising at least one network element according to claim 9.

11. Computer program, loadable into an internal memory of a digital processing unit, comprising software code portions adapted to control the steps according to any of the claims 1 to 7, when the computer program is executed on the digital processing unit.

12. Computer program according to claim 11, wherein the computer program is stored on a computer-readable medium.

Fig. 1 (Prior art)

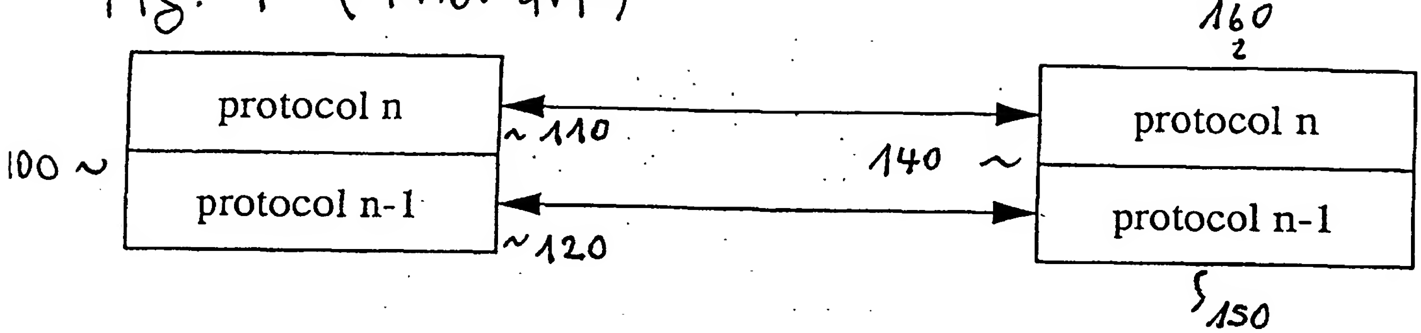


Fig. 2)

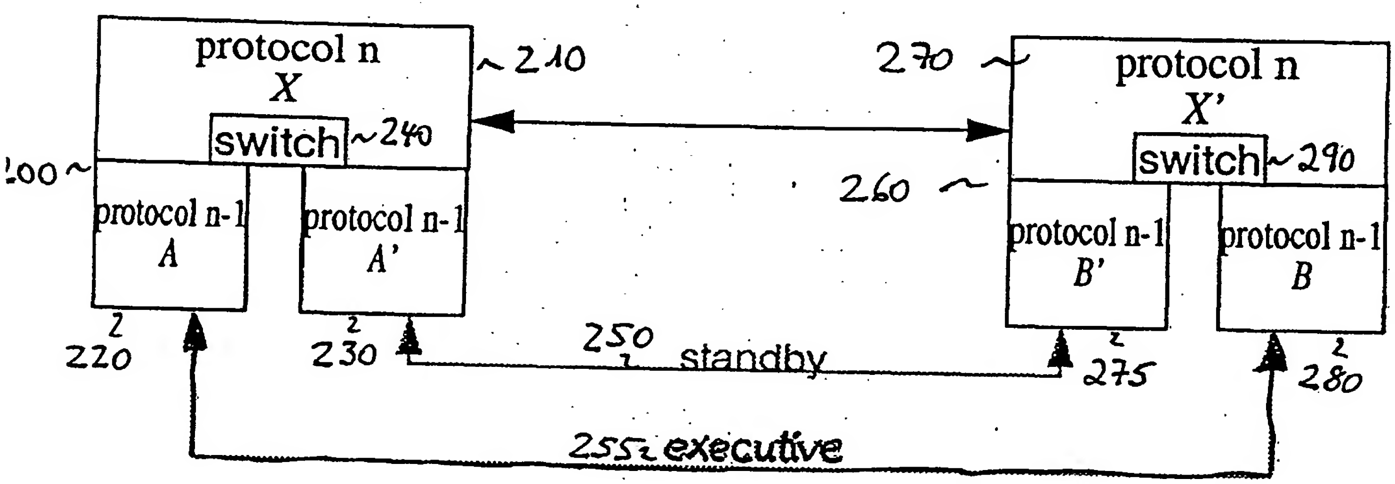


Fig. 3

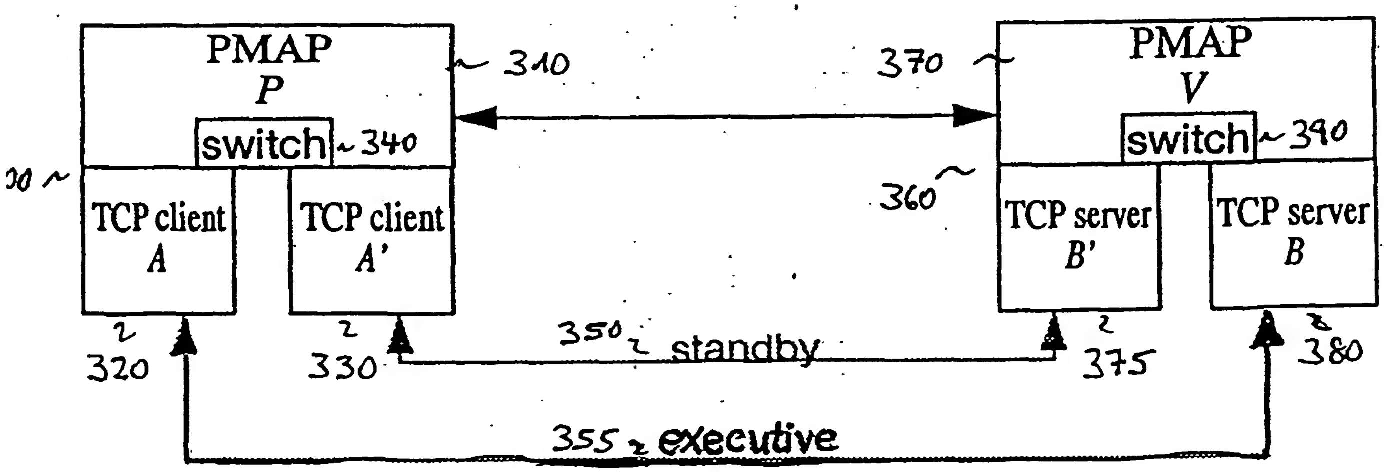


Fig. 4

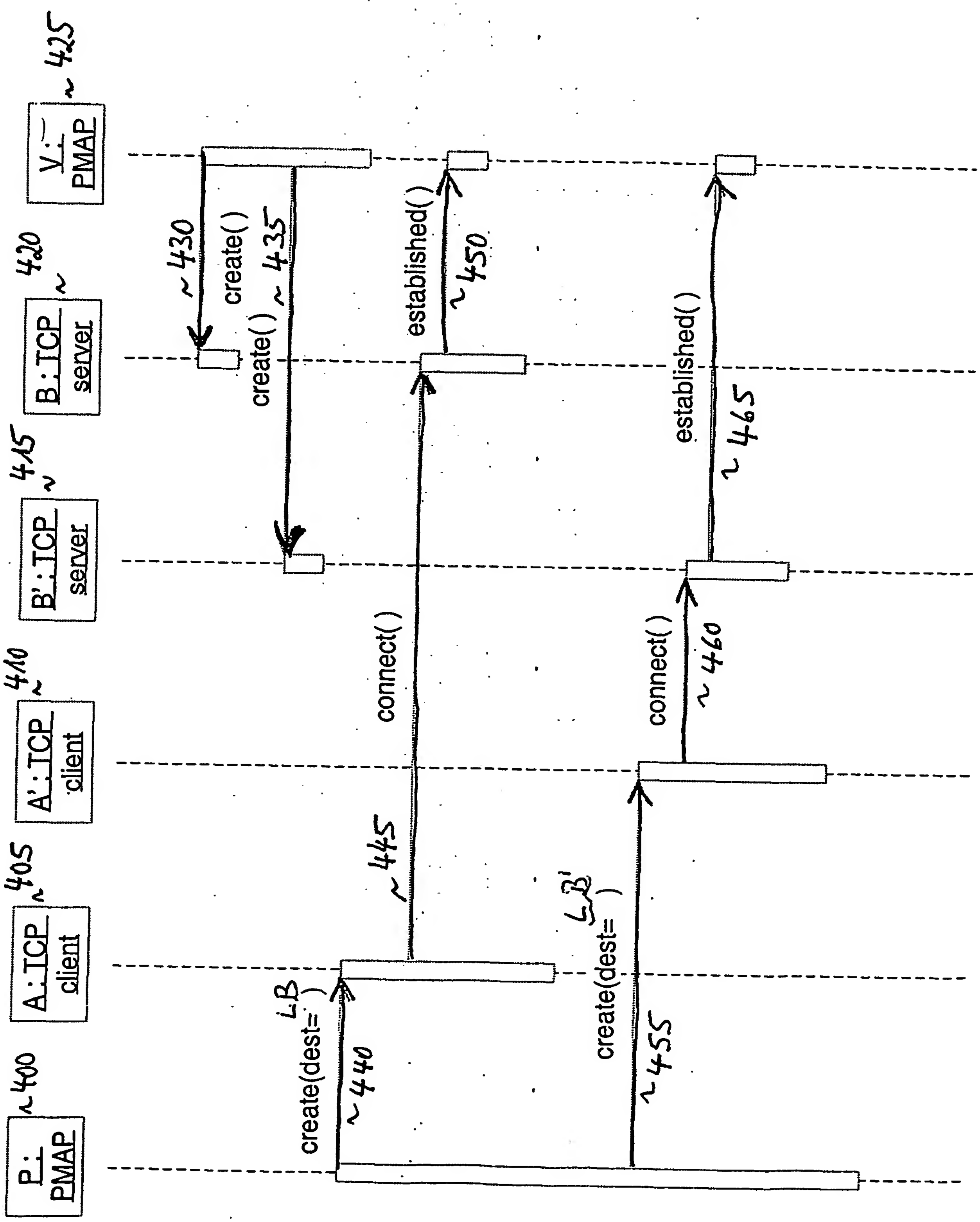
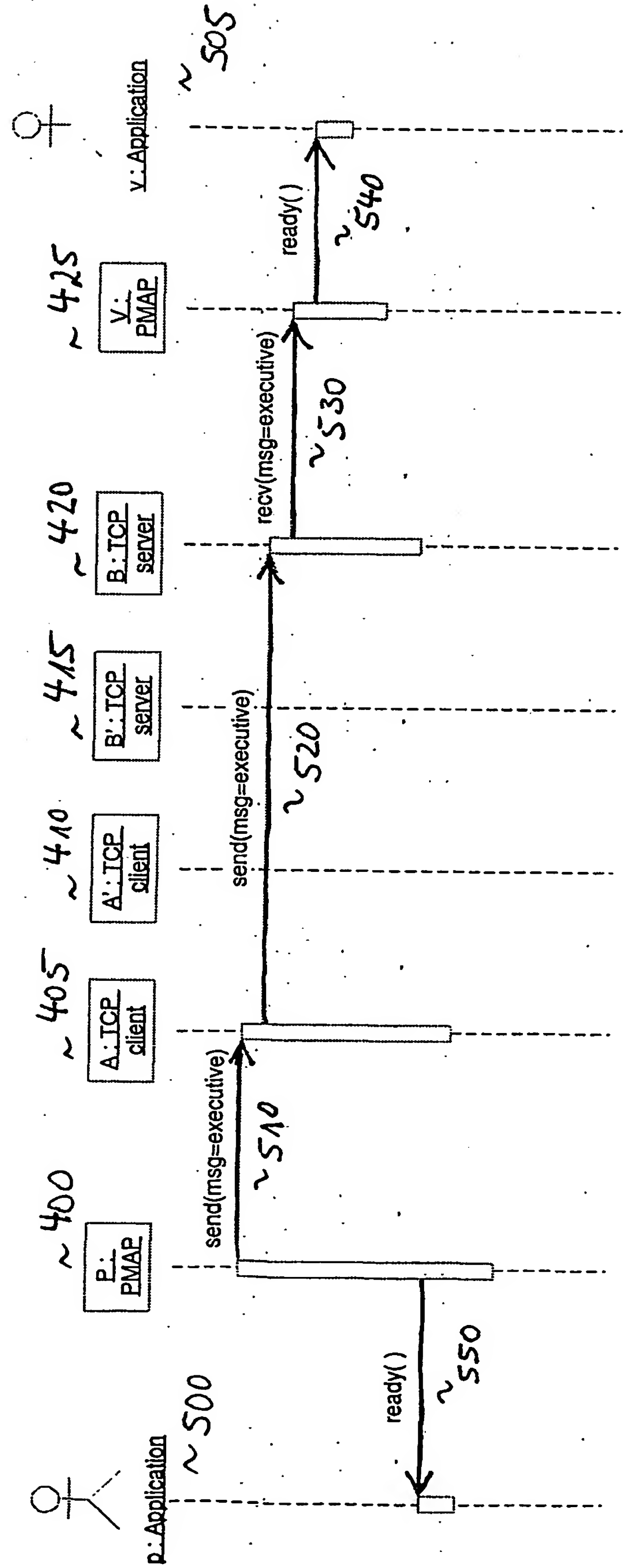

P: PMAP
400
A: TCP client
405
A': TCP client
410
B': TCP server
415
B: TCP server
420
V: PMAP
425
create()
430
create()
435
create(dest=LB)
440
connect()
445
established()
450
create(dest=LB')
455
connect()
460
established()
465

Fig. 5



v : Application
~505
v : PMAP
~425
B : TCP server
~420
B' : TCP server
~415
A' : TCP client
~410
A : TCP client
~405
p : PMAP
~400
p : Application
~500
ready( )
~540
recv(msg=executive)
~530
send(msg=executive)
~520
send(msg=executive)
~510
ready( )
~550

Fig. 6

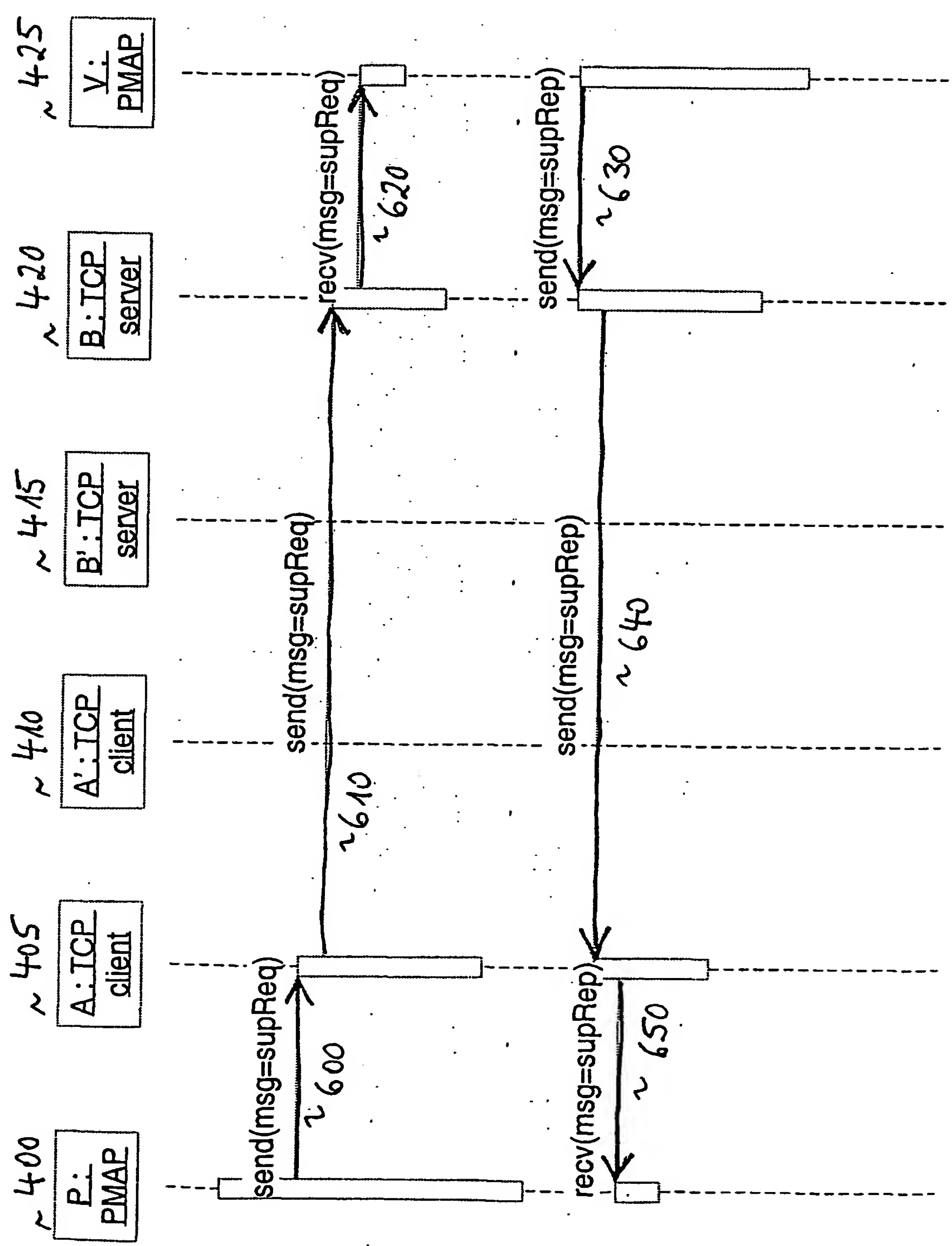
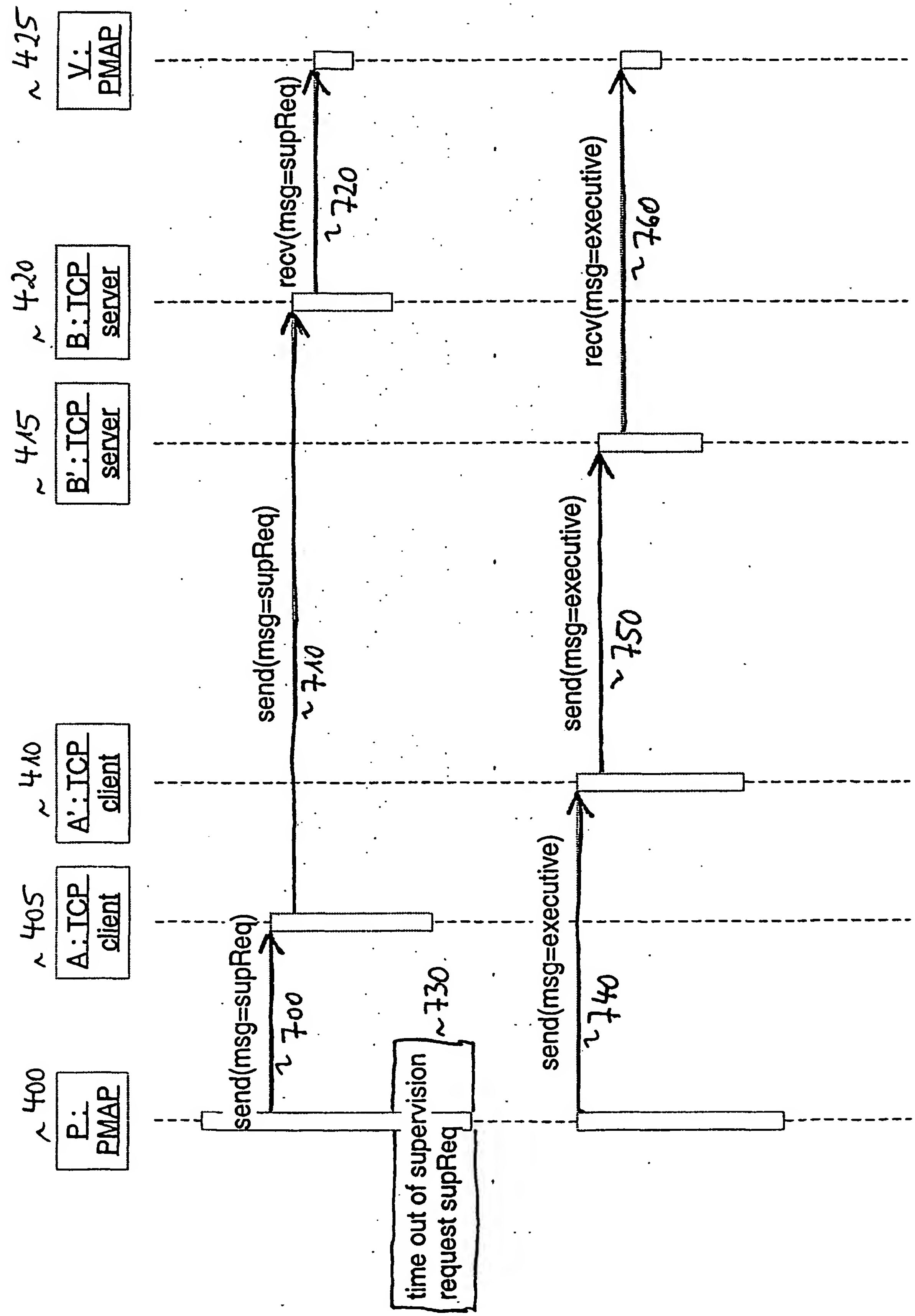

P: PMAP
400
A: TCP client
405
A': TCP client
410
B': TCP server
415
B: TCP server
420
V: PMAP
425
send(msg=supReq)
600
send(msg=supReq)
610
recv(msg=supReq)
620
send(msg=supRep)
630
send(msg=supRep)
640
recv(msg=supRep)
650

Fig. 7


400
P: PMAP
405
A: TCP client
410
A': TCP client
415
B': TCP server
420
B: TCP server
425
Y: PMAP
send(msg=supReq)
700
send(msg=supReq)
710
recv(msg=supReq)
720
time out of supervision request supReq
730
send(msg=executive)
740
send(msg=executive)
750
recv(msg=executive)
760